Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A data structure, comprising:

a head representing a first pointer to a first leaf node of a tree structure separate from the data structure;

a tail representing a second pointer to a second leaf node of the tree structure; and

a body, physically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes of the tree structure, wherein at least two contiguous empty nodes are maintained for the life of the date structure.

2. (Previously amended) The apparatus of claim 1, wherein the nodes of the tree structure comprise further comprising data of the same type.

3. (Previously amended) The apparatus of claim 1, wherein the nodes form tree structure is a sorted tree structure.

4. (Previously amended) The apparatus of claim 1, wherein the nodes of the tree structure are indexed.

5. (Previously amended) The apparatus of claim 1, wherein each of the first and second leaf node nodes comprises a number of data segments.

6. (Currently amended) A method for rapid insertion of data segments comprising:

providing a sorted tree structure;

preparing a redistribution data structure separate from the sorted tree structure, said redistribution data structure having

a head representing a first pointer to a first leaf node of the sorted tree structure,

a tail representing a second pointer to a second leaf node of the sorted tree structure, and

a body, logically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes of the sorted tree structure;

an inserting of a data segment into the sorted tree structure; and

a redistributing of the contiguous empty tree nodes by employing a the redistribution data structure, which enables to enable a more rapid insertion of the data segments,

wherein at least two contiguous empty nodes are maintained for the life of the data structure.

7. (Previously amended) The method of claim 6, wherein the data segments may be is inserted in any order.

8. (Previously amended) The method of claim 6, wherein the sorted tree structure comprises non-leaf and leaf nodes.

9. (Previously amended) The method of claim 6, wherein the tree nodes of the sorted tree structure are indexed.

10. (Previously amended) The method of claim 6, wherein each of the first and second leaf nodes comprises a number of data segments.

11. (Canceled)

12. (Previously amended) The method of claim 6, wherein the step of redistributing the contiguous empty nodes redistribution process comprises the data structure includes traversing the sorted tree structure in one of a first direction and a second direction.

13. (Previously added) The method of claim 12, wherein the first direction comprises a logical one and the second direction comprises a logical zero.

14. (Previously amended) The method of claim 12, wherein the data structure traverses the tree by step of traversing the sorted tree structure includes moving it's the head one leaf node towards its traveling one of the first and second directions.

15. (Previously amended) The method of claim 12, wherein the redistribution data structure traverses the tree structure in the first direction is towards non-decreasing indices.

16. (Previously amended) The method of claim 12, wherein the redistribution data structure traverses the tree structure in the second direction is towards non-increasing indices.

17. (Previously amended) The method of claim 6, wherein the redistribution data structure traverses the sorted tree structure when a the data segment is inserted and two conditions are met.

18. (Previously amended) The method of claim 17, wherein a first of the two conditions comprises a maximum threshold of filled spaces in the sorted tree structure, and a second of the two conditions comprises a minimum threshold of filled spaces in the sorted tree structure.

19. (Previously amended) The method of claim 17, wherein the two conditions are empirically determined.

20. (Previously amended) The method of claim 17, wherein the redistribution data structure traverses the sorted tree structure by moving one leaf node towards its traveling direction.

21. (Previously amended) The method of claim 20, wherein the head of the redistribution data structure further comprising comprises an empty leaf node.

22. (Previously amended) The method of claim 21, wherein certain conditions are met and the redistribution process step of redistributing the contiguous empty nodes continues.

23. (Previously amended) The method of claim 22, wherein the certain conditions are empirically calculated.

24. (Previously amended) The method of claim 21, wherein the redistribution process step of redistributing the contiguous empty nodes halts.

25. (Previously amended) The method of claim 24, wherein a data segment insertion restarts the redistribution process step of redistributing the contiguous empty nodes, and the traversal may continue from where it was last halted.

26. (Previously amended) The method of claim 20, wherein the head of the redistribution data structure comprises a non-empty leaf node.

27. (Previously amended) The method of claim 26, wherein the redistribution data structure copies the contents of it's the head into it's the tail.

28. (Previously added) The method of claim 26, wherein the redistribution data structure travels towards non-decreasing indices.

29. (Previously amended) The method of claim 28, wherein the sorted tree structure updates from leaf node level to root node level.

30. (Previously amended) The method of claim 29, wherein the contents of the head are cleared and the tail is moved a pre-calculated increment towards the traveling direction non-decreasing indices.

31. (Previously added) The method of claim 30, wherein the increment is empirically determined.

32. (Previously amended) The method of claim 30, wherein certain conditions are met and the redistribution process step of redistributing the contiguous empty nodes continues.

33. (Previously amended) The method of claim 32, wherein the certain conditions are empirically calculated.

34. (Previously amended) The method of claim 30, wherein the redistribution process step of redistributing the contiguous empty nodes halts.

35. (Previously amended) The method of claim 34, wherein a data segment insertion restarts the redistribution process step of redistributing the contiguous empty nodes, and the traversal may continue from where it was last halted.

36. (Previously added) The method of claim 27, wherein the redistribution data structure travels towards non-increasing indices.

37. (Previously amended) The method of claim 36, wherein the sorted tree structure updates between the tail and the nearest non-empty leaf node whose index is greater than the index of the tail.

38. (Previously amended) The method of claim 37, wherein the updates further comprising changes made the sorted tree structure updates from leaf node level to root node level.

39. (Previously amended) The method of claim 38, wherein the remainder of the sorted tree structure updates from root node level to leaf node level.

40. (Previously amended) The method of claim 39, wherein the contents of the head are cleared and the tail is moved a pre-calculated increment towards the traveling direction non-increasing indices.

41. (Previously added) The method of claim 40, wherein the increment is empirically determined.

42. (Previously amended) The method of claim 40, wherein certain conditions are met and the redistribution process step of redistributing the contiguous empty nodes continues.

43. (Previously amended) The method of claim 42, wherein the certain conditions are empirically calculated.

44. (Previously amended) The method of claim 40, wherein the redistribution process step of redistributing the contiguous empty nodes halts.

45. (Previously amended) The method of claim 44, wherein a data segment insertion restarts the redistribution process step of redistributing the contiguous empty nodes, and the traversal may continue from where it was last halted.

46. (Previously amended) The method of claim 6, wherein the sorted tree structure may be is reverse sorted.

47. (Previously amended) The method of claim 6, wherein the process step of redistributing the contiguous empty nodes maintains the invariants of a the sorted N-ary tree structure before and after the redistribution.

48. (Previously amended) The method of claim 6, wherein the redistribution process step of redistributing the contiguous empty nodes maintains a consistent lookup operation on the sorted tree structure.